

درخت عبارت^۱

هدف از این تمرین استفاده از مفاهیم وراثت و چندریختی در پیاده‌سازی یک درخت عبارت است. درخت عبارت از تعدادی عملگر و عملوند تشکیل شده‌است. هر یک از عملگرها دارای دو حالت کاری هستند (Arithmetic و Bitwise) و براساس اینکه در کدام حالت قرار دارند، یک عملیات خاص را مطابق با جدول زیر، روی عملوندها انجام می‌دهند.

حالت کاری عملگر	AND (x, y)	OR (x, y)	NOT (x)	XOR (x ₁ , ..., x _n)
Bitwise	x & y	x y	~ x	x ₁ ^ ... ^ x _n
Arithmetic	x * y	x + y	- x	Median (x ₁ , ..., x _n)

عملوندهای درخت نیز، مقادیر عددی‌ای هستند که در یکی از چهار مبنای ۲، ۸، ۱۰ یا ۱۶ در اختیار شما قرار می‌گیرند. در ابتدای برنامه ساختار درخت با فرمتی مشخص، در اختیار شما قرار می‌گیرد. هر یک از برگ‌های درخت را یک مقدار عددی (عملوند) و گره‌های داخلی درخت را نیز عملگرهای این تمرین در نظر بگیرید. عملگرها می‌توانند تک متغیره (NOT)، دو متغیره (AND, OR) و یا چند متغیره (XOR) باشند. لذا پس از ایجاد درخت عبارت، اطمینان حاصل کنید که محدودیت‌های مطرح‌شده برای همه ی گره‌های درخت رعایت شده باشند. به عنوان نمونه، **گره متناظر با یک عملگر دو متغیره، دقیقا دو فرزند داشته باشد و یا برگ‌های درخت، مقادیر عددی باشند.** به هر گونه خطای احتمالی با استفاده از Exception Handling رسیدگی کنید و در صورت نیاز پیام خطای مناسب را به کاربر نمایش دهید.

برای محاسبه حاصل یک درخت عبارت، از برگ‌ها حرکت می‌کنیم و عملگرها را در هر سطح اعمال می‌کنیم تا به ریشه درخت برسیم. مقدار مورد نظر برابر با حاصل عملگر متناظر با ریشه خواهد بود. جهت انعطاف‌پذیر بیشتر برنامه، سه قابلیت تغییر مبنای یک عملوند، حالت کاری یک عملگر و مقدار یک عملوند حین اجرای برنامه، پیاده‌سازی خواهند شد. در ادامه دستورهای مورد نیاز برای این کار را معرفی می‌کنیم:

محاسبه حاصل درخت

```
evaluate -<base>
```

پس از اجرای این دستور، حاصل درخت را در مبنای base نمایش می‌دهیم که base از بین مقادیر b, o, d, h انتخاب خواهد شد که به ترتیب برابر با مبناهای ۱۶، ۱۰، ۸، ۲ است.

تغییر مبنای یک عملوند خاص

```
rebase -<base> <index>
```

پس از اجرای این دستور، مقدار ذخیره شده در برگ با اندیس index را به مبنای base منتقل می‌کنیم. توجه کنید که این فرایند به معنی انتقال مقدار از یک مبنا به مبنای دیگر نیست، بلکه مقدار اولیه ثابت می‌ماند و تنها مبنا عوض می‌شود که در برخی سناریوها نیز امکان‌پذیر نخواهد بود. به طور مثال اگر مقدار عملوند 'a1' و مبنای آن ۱۶ باشد، امکان تغییر مبنا

¹ Expression Tree

به ۲ را نداریم چراکه در مبنای ۲ سمبل 'a' مفهومی ندارد. هنگام ساخت درخت نیز (در ابتدای برنامه)، مبنای پیش فرض همه عملوندها را دسیمال در نظر بگیرید.

تغییر حالت کاری یک عملگر خاص

```
remode -<mode> <index>
```

پس از اجرای این دستور، حالت کاری اپراتور متناظر با گره با اندیس index درخت را به حالت mode تغییر می دهیم. مقدار mode از بین حالت های a, b انتخاب می شوند که به ترتیب Arithmetic و Bitwise را مشخص می کنند. هنگام ساخت درخت نیز حالت پیش فرض همه عملگرها را Arithmetic در نظر بگیرید.

تغییر مقدار یک عملوند خاص

```
reset <index> <value>
```

توجه کنید که در پیاده سازی این تمرین، بایستی از مفاهیم وراثت و چندریختی استفاده کنید و به کاربردن دستورات شرطی مانند if یا switch برای تشخیص مبنای عملوند یا حالت عملگر حین انجام محاسبات مورد پذیرش قرار نخواهد گرفت!

ورودی

در خط اول ورودی استاندارد، عدد n (تعداد گره های درخت) وارد می شود و در n خط بعد نیز، اطلاعات مورد نیاز برای ایجاد هر گره درخت مطابق با الگوی زیر در اختیار شما قرار می گیرد:

```
<index> <parent_index> <type> <argument>
```

مقدار index، اندیس یک گره را مشخص می کند. توجه کنید که اندیس ریشه ی درخت برابر با صفر است. مقدار parent_index، اندیس پدر این گره در درخت را مشخص می کند. مقدار type یکی از دو حالت operator یا input است که به ترتیب عملگر و عملوند بودن یک گره را مشخص می کنند. argument نیز برای عملگرها، نوع عملگر و برای عملوندها مقدار آنها را مشخص می کند. خطوط بعدی ورودی استاندارد نیز، تعدادی فرمان مطابق با فرمت تعریف شده خواهند بود و برنامه شما به ترتیب به آنها رسیدگی خواهد کرد.

خروجی

با رسیدن به هر دستور evaluate در ورودی استاندارد، حاصل درخت را در یک خط، در خروجی استاندارد چاپ کنید.

خروجی نمونه	ورودی نمونه
25 1 -3 -100	6 0 -1 operator OR 1 0 operator AND 2 0 operator NOT 3 1 input 10 4 1 input 3 5 2 input 5 evaluate -d rebase -b 3 evaluate -d remode -b 1 evaluate -d reset 5 6 evaluate -b

دقت کنید

- برنامه‌ی شما باید در سیستم عامل لینوکس نوشته و با مترجم g++ ترجمه شود.
- در چاپ کردن خروجی نهایت دقت را به خرج دهید.